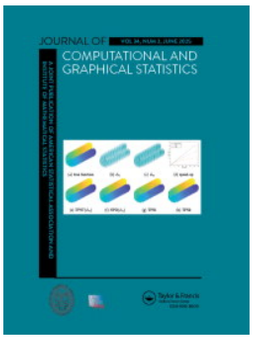

Journal of Computational and Graphical Statistics

ISSN: 1061-8600 (Print) 1537-2715 (Online) Journal homepage: www.tandfonline.com/journals/ucgs20

Efficient Approximation of Leverage Scores in Two-dimensional Autoregressive Models with Application to Image Anomaly Detection

Junlie Huang, Xinlai Kang, Qiannan Huang, Mengyu Li, Cheng Meng & Jingyi Zhang

To cite this article: Junlie Huang, Xinlai Kang, Qiannan Huang, Mengyu Li, Cheng Meng & Jingyi Zhang (19 May 2025): Efficient Approximation of Leverage Scores in Two-dimensional Autoregressive Models with Application to Image Anomaly Detection, Journal of Computational and Graphical Statistics, DOI: 10.1080/10618600.2025.2505732

To link to this article: https://doi.org/10.1080/10618600.2025.2505732

Efficient Approximation of Leverage Scores in Two-dimensional Autoregressive Models with Application to Image Anomaly Detection

Junlie Huang[a], Xinlai Kang[a], Qiannan Huang[a], Mengyu Li[a], Cheng Meng[b,*], Jingyi Zhang[c,†]

[a]Institute of Statistics and Big Data, Renmin University of China

[b]Center for Applied Statistics, Institute of Statistics and Big Data, Renmin University of China

[c]School of Science, Beijing University of Posts and Telecommunications

[†]Corresponding author, jyzhang2024@bupt.edu.cn

*Corresponding author, chengmeng@ruc.edu.cn

Abstract

Leverage scores quantify the influence of individual data points within a dataset and are widely used in subsampling methods to obtain a representative subsample. Numerous algorithms have been proposed to efficiently approximate leverage scores, thereby reducing the time complexity in model parameter estimation. In this paper, we study leverage scores in two-dimensional autoregressive models. We develop an efficient algorithm that accelerates the calculation of leverage scores by exploiting the unique structure of the covariate matrix specific to this model. Theoretically, we show that leverage scores can be approximated quickly and accurately by deriving an error bound between the approximated and true values. Numerical studies on synthetic datasets demonstrate the superior performance of the proposed algorithm. Additionally, when applying leverage scores in the two-dimensional autoregressive model to anomaly detection tasks, we achieve competitive detection results compared to state-of-the-art methods, with significantly reduced computational time. Furthermore, the efficient approximation of the leverage scores further reduces the time cost without loss of detection accuracy.

Keywords: Order selection; Ordinary least squares (OLS); Parameter estimation; Subsampling

1 Introduction

A two-dimensional (2D) autoregressive (AR) model predicts each pixel value in an image based on its surrounding pixels in both horizontal and vertical directions. 2D AR models have extensive applications in signal and image processing. For example, they have played a predominant role in texture analysis (Köppel et al., 2015; Oe, 1993; Lu and Xu, 1995), image restoration (Kokaram, 2004; Kaufman and Tekalp, 1991), signal compression (Zhang and Liang, 2017), cancer detection and classification (Zielinski et al., 2010), among others.

Parameter estimation is a common challenge when applying 2D AR models. Several methods have been developed to address this, with the ordinary least squares (OLS) estimator and the Yule-Walker estimator being the most commonly used. Both estimators are consistent and asymptotically effective when the model structure is known and accurate. However, in scenarios where the data exhibit non-stationarity or when the true autocorrelation structure of the data is unknown, the OLS estimator is preferred over the Yule-Walker estimator, as it does not require strict stationarity assumptions and can more effectively handle data with varying autocorrelation structures.

Despite its effectiveness, the OLS estimator suffers from a huge computational burden, especially when dealing with large images or high model orders. Specifically, the computation cost of the traditional OLS method is of the order $O(mnp^2q^2)$, where $m \times n$ is the image size and p, q are the orders of the 2D AR model in horizontal and vertical directions, respectively. If we let $N = mn$ represent the number of pixels and $d = O(pq)$ represent the number of parameters, the time complexity can be written as $O(Nd^2)$.

The subsampling method is a powerful technique that can efficiently reduce this computational burden. The application of the subsampling method in large-scale data analysis has been widely explored in various fields, including linear regression (Drineas et al., 2011, 2012; Ma et al., 2015; Ma and Sun, 2015; Meng et al., 2017; Wang et al., 2017; Zhang et al., 2018; Li and Meng, 2020; Ma et al., 2022; Li et al., 2024a), generalized linear regression (Wang et al., 2018; Ai et al., 2021; Jun Yu and Zhang, 2022), quantile regression (Ai et al., 2021), streaming time series (Xie et al., 2019, 2023), Gaussian mixture model (Feldman et al., 2011), nonparametric regression (Gu and Kim, 2002; Meng et al., 2020, 2022; Li et al., 2024b), optimal transport(Li et al., 2023a, b; Hu et al., 2025), among others.

In large-scale least squares regression, subsampling methods can be divided into optimal and randomized subsampling approaches (Li and Meng, 2020). Optimal subsampling methods select a deterministic subsample based on certain rules, such as A-, D- and E-optimality (Pukelsheim, 2006; Wang et al., 2019; Meng et al., 2021; Xie et al., 2023). Although these methods perform excellently in conventional linear regression, the unique structure of the 2D AR model limits their performance in the corresponding OLS estimation. On the other hand, randomized subsampling methods, which select subsamples based on data-dependent nonuniform sampling probabilities, such as the leverage subsampling method (Drineas et al., 2006; Ma et al., 2015; Meng et al., 2017), have been shown to outperform simple random subsampling in numerous studies (Drineas et al., 2006; Drineas et al., 2012; Ma et al., 2015). Random projection-based techniques developed by Drineas et al. (2012) can efficiently approximate leverage scores, reducing the time cost from $O(Nd^2)$ to $O(Nd\log N + d^3)$ in conventional linear regression. Additionally, in specific situations, unique model structures can be utilized to further accelerate computation. For instance, in the 1D AR model, the

covariates matrix $X_{N,d}$ possesses the Toeplitz structure, which Eshragh et al. (2022) used to develop a leverage-based iterative algorithm that approximates the OLS estimator of the 1D AR model more efficiently, decreasing the time complexity to $O(Nd + d^4 \log d)$. For more complex multidimensional streaming time series, Xie et al. (2023) proposed the Relaxed-LSS method, which substantially accelerates parameter estimation in VAR models and reduces the time complexity to $O\left((qN + c)d^2\right)$ with $qN + c \ll N$.

In this paper, we propose a novel subsampling method for parameter estimation in the 2D AR model, reducing the average time complexity to $O(mnpq + (pq)^3 \log(pq))$, or $O(Nd + d^3 \log d)$ equivalently. To our knowledge, this is the first leverage score subsampling algorithm designed for large-scale 2D AR models. Furthermore, we develop a novel methodology that incorporates the 2D AR model framework for image analysis. Notably, in anomaly detection tasks, our approach demonstrates superior performance compared to current state-of-the-art techniques, while maintaining significantly improved computational efficiency.

Methodologically, we first develop an iterative formula that can exactly calculate leverage scores of the 2D AR model with different orders. In each iteration, leverage scores are updated by solving the OLS problem for a sub-model. Due to the unique structure of the 2D AR model, leverage scores from the previous iteration can be used to approximate the OLS problem in the current iteration. Furthermore, we incorporate the leverage-based subsampling method and reduce the computational cost from $O(Nd^2)$ to $O(Nd + d^3 \log d)$ per iteration. After obtaining leverage scores, parameters can be estimated using the same subsampling technique. Theoretically, we show that the resulting subsample-based estimator closely approximates the exact OLS estimator with high probability.

When the true order of the 2D AR model is unknown, we can fit a series of 2D AR models using leverage scores obtained during the iterative process and select the optimal 2D AR model based on some criteria, such as the minimal mean square error (MSE). Specifically, the proposed method yields a total of d 2D AR models, with the number of parameters ranging from 1 to d. The time complexity for fitting these models is $O(Nd^2 + d^4 \log d)$, with an average time cost of $O(Nd + d^3 \log d)$ per model. In contrast, the OLS method and the conventional order selection method have a total time complexity of $O(Nd^3)$ and an average time cost of $O(Nd^2)$ per model, respectively. Extensive simulations show that the proposed method yields superior performance compared to other mainstream competitors.

We demonstrate the effectiveness of the proposed method by applying it to a real-world anomaly detection task, which has extensive application in object detection (Jang et al., 2023), mineral surveys (Sun et al., 2023), military surveillance (Palm et al., 2022), and precision agriculture (Kang et al., 2017). We propose an innovative approach that utilizes standardized leverage scores derived from the 2D AR model as anomaly scores. The anomaly pixels are then detected by identifying those with leverage scores exceeding a certain threshold. Empirical results show that anomaly pixels can be efficiently identified using leverage scores of the 2D AR model. We evaluate anomaly detection performance using the area under the receiver operating characteristic curve (AUC). The results show that the leverage scores of the 2D AR model achieve the same detection accuracy as state-of-the-art anomaly detection methods while requiring much less computational time. In addition, the

proposed subsampling method can be applied to further accelerate the calculation of leverage scores while maintaining nearly identical accuracy.

A related application is facial skin analysis using hyperspectral images. Different hyperspectral wavelengths of the skin encompass diverse layers of skin information (Seroul et al., 2016; Gevaux et al., 2021). These diverse pieces of information can be utilized to extract characteristics of interest, such as oxygen saturation, blood volume fraction, and melanin concentration (Seroul et al., 2016; Gevaux et al., 2019). We propose identifying regions of skin lesions through leverage scores of the 2D AR model. The results show that our method can efficiently distinguish pigmentation and acne on the skin using the wavelength band around 900 nm.

The remainder of this paper is organized as follows. We start in Section 2 by introducing the preliminaries of the 2D AR model and leverage scores. In Section 3, we develop the iterative formula and provide the details of the main algorithm. The theoretical properties of the proposed estimators and the time complexity are presented in Section 4. We examine the performance of the proposed method through extensive synthetic datasets in Section 5. Real-data analysis is provided in Section 6. Extensions, technical details, and proofs of theorems are relegated to the Appendix.

2 Preliminaries

In this paper, vectors and matrices are denoted by bold lower-case and bold upper-case letters (e.g., $\mathbf{x}$ and $\mathbf{X}$), respectively. All vectors are assumed to be column vectors. Three-dimensional tensors are represented by calligraphic letters (e.g., $\mathcal{X}$). The third dimension of tensors is referred to as a channel, that is, $\mathcal{X} \in \mathbb{R}^{m \times n \times c}$ consists of c channels, each of which is a matrix of size $m \times n$. The condition number of the tensor $\mathcal{X}$ is defined as $\kappa(\mathcal{X}) \coloneqq \kappa(\mathbf{X})$, where $\mathbf{X} \in \mathbb{R}^{mn \times c}$ is constructed such that $\mathbf{X}_{in+j-n,k} = \mathcal{X}(i, j, k)$, and $\kappa(\mathbf{X})$ is the condition number of the matrix $\mathbf{X}$. Table 1 lists the tensor and matrix operations used throughout this paper.

In short, we use $\star$ to represent the element-wise product between matrix and tensor along the first two dimensions. The result of $\circ$ is a matrix whose (i, j)th element is the Frobenius inner product of the ith channel of the front tensor and the jth channel of the behind tensor. The result of $\odot$ is a matrix whose (i, j)th element is the inner product of the behind vector and the (i, j)th vector of the front tensor. Here we regard a matrix as a tensor whose third dimension is 1 and a vector as a matrix whose second dimension is 1.

2.1 Two-dimensional autoregressive model

A two-dimensional time series $\{y_{i,j} \mid i = 0, \pm 1, \pm 2, \ldots; j = 0, \pm 1, \pm 2, \ldots\}$ is called (weakly) stationary, if the mean $\mathbb{E}[y_{i,j}]$ is independent of subscripts i, j, and the auto-covariance $\mathrm{Cov}(y_{i,j}, y_{i+u,j+v})$ depends only on the lags u, v for any integers i, j, u, v.

Definition 1 (2D AR Model).

A stationary two-dimensional time series $\left\{ y_{i,j} \mid i = 0, \pm 1, \ \pm 2, \ldots; j = 0, \pm 1, \pm 2, \ldots \right\}$ *follows a 2D AR model with order* (p, q) *, denoted by 2D AR* (p, q) *, if*

$$y_{i,j} = \sum_{(u,v) \in L} \phi_{u,v} y_{i+u, j+v} + \varepsilon_{i,j}, \quad (1)$$

$$L = \{-p, -p+1, \ldots, p\} \times \{-q, -q+1, \ldots, q\} / \{(0,0)\},$$

where (u, v) *denotes the plane offset between response and predictors, L is the support set, and* $\{\varepsilon_{i,j}\}$ *is mutually uncorrelated Gaussian white noise with mean zero and constant variance* σ^2 *.*

Definition 1 indicates that each point $y_{i,j}$ is influenced by the points within a $(2p+1) \times (2q+1)$ rectangle centered at $y_{i,j}$. Parameters $\{\phi_{u,v}\}$ quantify the influence at each location within this rectangle. While the support region is rectangular by default, it can be customized by selecting an appropriate support set *L*. One can refer to Choi and Politis (2007) for various support regions, including causal and non-causal 2D AR models.

Each 2D AR (p, q) model contains $(2p+1)(2q+1)+2$ unknown parameters: the order (p, q), $(2p+1)(2q+1)-1$ coefficients $\{\phi_{u,v}\}$, and the variance of the white noise σ^2. Among these, the coefficients $\{\phi_{u,v}\}$ are the most critical, as they directly determine the structure of the 2D AR model. Therefore, we first focus on estimating the coefficients $\{\phi_{u,v}\}$ through an efficient approximation of leverage scores. Later, we will discuss how the order (p, q) can be estimated along with the leverage score approximation.

Let $\{y_{i,j} \mid i = 1, 2, \ldots, m; j = 1, 2, \ldots, n\}$ be a realization of the 2D AR (p, q) model, where the order (p, q) is known. The relationship given in equation (1) holds for $i = p+1, \ldots, m-p; j = q+1, \ldots, n-q$. The equation (1) can be represented in vector form as $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e}$ by arranging the response and predictors in a specific order (e.g., row-first or column-first). In this way, the OLS estimator of $\boldsymbol{\beta}$, i.e., the concatenated form of $\{\phi_{u,v}\}$, can be derived in the same way as linear regression. Nonetheless, this method destroys the unique spatial structure of the 2D AR model, which can be crucial for enhancing the efficiency of leverage score approximation and parameter estimation.

To fully exploit the unique structure of the 2D AR model, we reformulate it in tensor form as

$$\mathbf{Y} = \mathcal{X} \odot \boldsymbol{\beta} + \mathbf{E}, \quad (2)$$

where $\mathbf{Y} \in \mathbb{R}^{(m-2p) \times (n-2q)}$ with $Y_{i,j} = y_{i+p, j+q}$, $\mathbf{E} \in \mathbb{R}^{(m-2p) \times (n-2q)}$ with $E_{i,j} = \varepsilon_{i+p, j+q}$, $\mathcal{X} \in \mathbb{R}^{(m-2p) \times (n-2q) \times |L|}$ with $\mathcal{X}(:,:,k) = \{\mathbf{X} \in \mathbb{R}^{(m-2q) \times (n-2q)} \mid X_{i,j} = Y_{i+u_k, j+v_k}\}$, and $\boldsymbol{\beta} \in \mathbb{R}^{|L|}$ with $\beta_k = \phi_{u_k, v_k}$. Tensor calculus facilitates the derivation of efficient leverage score approximations and provides a more intuitive understanding. The OLS estimator of (2) can be

obtained either by rewriting the result from multiple linear regression into tensor form or by minimizing the sum of squared residuals, that is,

$$\boldsymbol{\beta} = (\mathcal{X} \circ \mathcal{X})^{-1} (\mathcal{X} \circ \mathbf{Y}). \quad (3)$$

The visualization of the formula is provided in Figure 1.

A significant extension of the autoregressive (AR) model is the vector autoregressive (VAR) model, which generalizes the scalar operations in AR models to vector spaces, providing a powerful framework for analyzing multidimensional time series (Xie et al., 2019, 2023). Although the VAR model can be expressed in a tensor form analogous to Equation (2), it differs from the 2D AR model in two key respects. First, while the response in the VAR model can be written as a matrix, only the time dimension is extendable, with the other dimension fixed as the length of the vector. In contrast, the 2D AR model has two time dimensions, both of which can be extended arbitrarily. Second, the parameter for each lag is a matrix in the VAR model but a scalar in the 2D AR model. After reparameterizing to tensor form, elements in the response matrix of the 2D AR model share the same parameters, whereas in the VAR model, elements in different rows are associated with different row vectors of the parameter matrix.

2.2 Leverage score

Leverage score measures the impact of individual data points in a dataset. In 1D models, the leverage score of the *i*th data is defined as $l(i) = \partial \hat{y}_i / \partial y_i$ (Eshragh et al., 2022). Specifically, in a linear model, we have $l(i) = H_{i,i}$, where $H_{i,i}$ is the *i*th diagonal element of the hat matrix $\mathbf{H} = \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$.

Leverage scores are widely used in subsampling methods to acquire representative subsamples, particularly when dealing with large-scale datasets (Ma et al., 2015, 2022; Derezinski and Warmuth, 2017; Derezinski et al., 2018). By subsampling according to leverage scores, points are selected with probabilities proportional to their influence, leading to more accurate estimates than random sampling. This approach significantly reduces the data size, making the computation cost affordable while maintaining reliable parameter estimation.

However, calculating exact leverage scores has the same time complexity as estimating parameters on the entire dataset. The high computational cost makes it infeasible to use exact leverage scores to accelerate parameter estimation. Therefore, we propose a method to approximate the leverage scores more efficiently, thereby reducing the overall time complexity of parameter estimation. The details will be discussed in the next section.

For a 2D AR model, the leverage score at position (i, j) can be defined as $l(i, j) = \partial \hat{Y}_{i,j} / \partial Y_{i,j}$. Given that $\mathbf{Y} = \mathcal{X} \odot \boldsymbol{\beta}$, combined with the formula of $\boldsymbol{\beta}$ in (3), we can express the leverage score as

$$l(i, j) = \mathcal{X}(i, j, :)^T (\mathcal{X} \circ \mathcal{X})^{-1} \mathcal{X}(i, j, :), \quad (4)$$

where $\mathcal{X}(i, j,:)$ represents the column vector along the third dimension corresponding to the position (i, j).

3 Main algorithm

In this section, we exploit the unique structure of the data tensor induced by the 2D AR model to derive the exact recursive formulation of leverage scores, and then develop a fast algorithm to approximate leverage scores efficiently.

3.1 Exact leverage scores

We first introduce some notations in Definition 2.

Definition 2 (Sub-model).

For $t = 1,\ldots,|L|$, *we define the sub-model* $\mathbf{Y} = \mathcal{X}_{1:t} \odot \boldsymbol{\beta} + \mathbf{E}$, *where* $\mathcal{X}_{1:t} \in \mathbb{R}^{(m-2p)\times(n-2q)\times t}$ *represents a sub-tensor consisting of the first t channels of* $\mathcal{X}$, *and* $\mathcal{X}_t \in \mathbb{R}^{(m-2p)\times(n-2q)}$ *represents the t-th channel of* $\mathcal{X}$. *The leverage score* $l_t(i, j)$ *(or its normalized form* $\pi_t(i, j)$*) for the* (i, j)*th point in this sub-model is defined as*

$$\begin{aligned} l_t(i, j) &= \mathcal{X}_{1:t}(i, j,:)^T (\mathcal{X}_{1:t} \circ \mathcal{X}_{1:t})^{-1} \mathcal{X}_{1:t}(i, j,:), \\ \pi_t(i, j) &= \frac{l_t(i, j)}{t}, \end{aligned}$$

for $i = p+1,\ldots,m-p;\ j = q+1,\ldots,n-q$.

The following proposition showcases how the leverage scores associated with a 2D AR(p,q) model can be recursively computed using those arising from a sub-model.

Proposition 1 (Exact Leverage Score Computations).

The leverage scores for a 2D AR (p,q) *model can be computed recursively as follows:*

$$l_1(i, j) = \frac{\mathcal{X}(i, j,1)^2}{(\mathcal{X}(:,:,1)\|_F^2}, \quad (5)$$

$$l_t(i, j) = l_{t-1}(i, j) + \frac{\mathbf{R}_{t-1}(i, j)^2}{(\mathbf{R}_{t-1}\|_F^2}, \quad \text{for } t = 2,\ldots,|L|, \quad (6)$$

where $\mathbf{R}_t$ *is the regression residual of recursive model* $\mathcal{X}_t = \mathcal{X}_{1:(t-1)} \odot \boldsymbol{\beta} + \mathbf{E}$. *Specifically,*

$$\boldsymbol{\beta}_{t-1} = (\mathcal{X}_{1:(t-1)} \circ \mathcal{X}_{1:(t-1)})^{-1} (\mathcal{X}_{1:(t-1)} \circ \mathcal{X}_t), \quad (7)$$

$$\mathbf{R}_{t-1} = \mathcal{X}_t - \mathcal{X}_{1:(t-1)} \odot \boldsymbol{\beta}_{t-1}. \quad (8)$$

Proposition 1 shows that leverage scores of the 2D AR model (2) can be exactly calculated through recursive updates (6) applied on tensor channels, starting from the initial condition (5). At each iteration, we regress the tth channel of $\mathcal{X}$ on its previous $(t-1)$ channels and update the leverage scores using the regression residuals through (6).

Although Proposition 1 is appealing at first glance, two major challenges render it impractical: the need for exact leverage scores of sub-models, and the requirement of exact residuals from corresponding OLS estimations. In the context of big data, calculating either exactly would negate the benefits of subsampling.

To address these issues, we first approximate the regression residuals through subsampling and then use these approximations to estimate the corresponding leverage scores. This approach allows us to rapidly obtain approximate leverage scores, with guaranteed relative error bounds, as discussed in Section 4.

3.2 Approximate leverage scores

One crucial property of leverage scores in AR models is their independence from the response value, as shown in (4). This property ensures that leverage scores of the recursive model $\mathcal{X}_t = \mathcal{X}_{1:(t-1)} \odot \boldsymbol{\beta} + \mathbf{E}$ are identical to those of the sub-model $\mathbf{Y} = \mathcal{X}_{1:(t-1)} \odot \boldsymbol{\beta} + \mathbf{E}$. Therefore, $l_{t-1}(i, j)$ is exactly the leverage score of the recursive model $\mathcal{X}_t = \mathcal{X}_{1:(t-1)} \odot \boldsymbol{\beta} + \mathbf{E}$.

Residual approximation. To approximate the residuals in (8), one approach is to subsample the predictor tensor $\mathcal{X}_{1:(t-1)}$ and solve the corresponding reduced OLS problem. Specifically, we consider the masked data tensor

$$\mathcal{X}_{1:(t-1)} = \mathbf{S} \star \mathcal{X}_{1:(t-1)},$$

where $\mathbf{S} \in \mathbb{R}^{m \times n}$ is a mask matrix with s elements equal to 1 and the others equal to 0. These s nonzero elements are chosen randomly without replacement according to probabilities $\{\pi_{t-1}(i, j) = l_{t-1}(i, j)/(t-1)\}$. Similar to the sampling matrix in basic leverage subsampling, mask $\mathbf{S}$ serves the same purpose to represent the indexes of selected samples here. Using $\mathcal{X}_{1:(t-1)}$, the approximated parameter $\boldsymbol{\beta}_{t-1}$ is calculated as

$$\boldsymbol{\beta}_{t-1} = (\mathcal{X}_{1:(t-1)} \circ \mathcal{X}_{1:(t-1)})^{-1} (\mathcal{X}_{1:(t-1)} \circ \mathcal{X}_t), \quad (9)$$

where $\mathcal{X}_t = \mathbf{S} \star \mathcal{X}_t$. Figure 2 visualizes the process of subsampling and parameter approximation. The residuals are then approximated by

$$\mathbf{R}_{t-1} = \mathcal{X}_t - \mathcal{X}_{1:(t-1)} \odot \boldsymbol{\beta}_{t-1}. \quad (10)$$

Leverage score approximation. Then, we can replace the exact residuals in (6) by the approximated counterparts (10) to approximate the leverage scores. Moreover, in the next iteration, we can continue to subsample the data tensor and response matrix using the updated leverage scores, thus generating further approximated residuals. In this way, beginning with

an initial leverage score l_1, we can recursively compute the approximations of leverage scores $\hat{l}_2,\ldots,\hat{l}_{|L|}$. The formal process is described in Proposition 2.

Proposition 2 (Two-dimensional Fully-approximate Leverage Scores).

For an 2D AR (p,q) *model with* $p,q>0$ *, the fully-approximate leverage scores are defined by the following equation*

$$\hat{l}_t(i,j) := \begin{cases} l_1(i,j), & \text{for } t=1, \\ \hat{l}_{t-1}(i,j) + \dfrac{\mathbf{R}_{t-1}(i,j)^2}{\| \mathbf{R}_{t-1} \|_F^2}, & \text{for } t \geq 2, \end{cases} \quad (11)$$

where

$$\mathbf{R}_{t-1} := \mathcal{X}_t - \mathcal{X}_{1:(t-1)} \odot \boldsymbol{\beta}_{t-1}, \quad (12)$$

$$\boldsymbol{\beta}_{t-1} := (\mathcal{X}_{1:(t-1)} \circ \mathcal{X}_{1:(t-1)})^{-1} (\mathcal{X}_{1:(t-1)} \circ \mathcal{X}_t), \quad (13)$$

and $\mathcal{X}_{1:(t-1)}$ *and* $\mathcal{X}_t$ *are respectively the reduced data tensor and response matrix sampled from the recursive model* $\mathcal{X}_t = \mathcal{X}_{1:(t-1)} \odot \boldsymbol{\beta} + \mathbf{E}$*, according to the probability*

$$\hat{\pi}_{t-1}(i,j) = \frac{\hat{l}_{t-1}(i,j)}{t-1} \quad \text{for } i = p+1,\ldots,m-p;\ j = q+1,\ldots,n-q. \quad (14)$$

Based on the above discussion, we introduce the 2D AR leverage score approximation algorithm, outlined in Algorithm 1. To our knowledge, this is the first leverage score subsampling algorithm designed for large-scale 2D AR models. The theoretical results and some details of Algorithm 1 will be discussed in the next section. The support set L is provided in Algorithm 1, and the procedure for order selection—when the order is unknown—is comprehensively detailed in the "Order Selection" section of the Appendix.

Algorithm 1 Efficient 2D AR leverage score approximation

Input:

- Two-dimensional data $\mathbf{Y}$;

- Model order (p,q) and support set *L*;

- Constant parameters $0<\varepsilon<1$, and $0<\delta_0<1$.

Step 1. Construct the predictor tensor $\mathcal{X}$.

Step 2. Calculate the initial leverage score l_1. Set $\hat{l}_1 = l_1$ and $t=2$.

while $t \leq |L|$ **do**

Step 3. Set $\delta = \delta_0 / |L|$ and $h \in 1 - O(t\sqrt{\varepsilon})$. Set $s \in O(t \log(t/\delta)/(h\varepsilon^2))$.

Step 4. Subsample $\mathcal{X}_{1:(t-1)}$ and $\mathcal{X}_t$ along the first two dimension with probabilities $\{\hat{\pi}_{t-1}\}$ defined by (14) and subsample size *s*, resulting in the reduced data tensor $\mathcal{X}_{1:(t-1)}$ and response matrix $\mathcal{X}_t$.

Step 5. Compute the approximated OLS estimator $\boldsymbol{\beta}_{t-1}$ according to (13).

Step 6. Update leverage scores as in (11) to get $\hat{l}_t$.

Step 7. $t \leftarrow t+1$

end while

Step 8. Compute the approximated OLS estimator $\boldsymbol{\beta}_{|L|}$ according to (13) by replacing $\mathcal{X}_t$ with $\boldsymbol{Y}$.

Output: Leverage scores $\hat{l}_{|L|}$; estimated parameters $\boldsymbol{\beta}_{|L|}$.

4 Theoretical results

In this section, we study the theoretical properties of the leverage scores obtained from Algorithm 1. We show that our approximations of leverage scores possess relative error bounds with high probability. Additionally, we discuss the computational complexity of Algorithm 1, showing that it can fit a 2D AR model with an average computational complexity of $O(Nd + d^3 \log(d))$, where *N* is the number of pixels and *d* is the number of parameters. All proofs of this section are presented in the Appendix.

We derive the relative error bounds for individual leverage scores and estimated parameters obtained in Algorithm 1. The results are summarized in Theorem 1 below.

Theorem 1 (Relative Errors for Fully-approximate Leverage Scores).

For the two-dimensional fully-approximation leverage scores and estimated parameters derived from Algorithm 1, we have with probability at least $1-\delta_0$,

$$\frac{|\hat{l}_t(i,j) - l_t(i,j)|}{l_t(i,j)} \le (1+3\eta_{t-1}\kappa^2(\mathcal{X}_{1:t}))(t-1)\sqrt{\varepsilon}, \quad (15)$$

$$\left\|\boldsymbol{\beta}_t - \boldsymbol{\beta}_t\right\| \le \sqrt{\varepsilon}\eta_t \left\|\boldsymbol{\beta}_t\right\|, \quad (16)$$

where

$$\eta_t = \kappa(\mathcal{X}_{1:t})\sqrt{\xi^{-2}-1}, \quad (17)$$

$\kappa(\mathcal{X}_{1:t})$ *is the condition number of tensor* $\mathcal{X}_{1:t}$, *and* $\xi \in (0,1]$ *is the fraction that* $\mathcal{X}_{t+1}$ *can be represented by* $\mathcal{X}_{1:t}$, *that is,* $\xi := (\mathcal{X}_{1:t} \odot \boldsymbol{\beta}_t \,(_F / (\mathcal{X}_{t+1} \,(_F$.

Theorem 1 shows that the relative errors of fully-approximation leverage scores and estimated parameters from Algorithm 1 can be bounded with high probability. Besides, Theorem 1 implies the misestimation factor *h* (cf. Lemma 1 in Supplementary Materials) for the two-dimensional fully-approximation leverage scores is $1-O(t\varepsilon)$. To obtain an overall success probability of $1-\delta_0$, the failure probability of each iteration is set as $\delta=1-\sqrt[|L|]{1-\delta_0}\sim O(\delta_0/|L|)$. Combining Lemma 1 with this result leads to the choice of subsample size *s* in each iteration, as Step 3 in Algorithm 1.

Theorem 2 gives the computational complexity of Algorithm 1.

Theorem 2 .

For an input 2D AR(p,q) *model, let* $d=|L|$ *represents the number of parameters and* $N=mn$ *represents the number of pixels. The total time complexity of Algorithm 1 to fit d models is* $O\left(Nd^2+d^4\log d/\varepsilon^2\right)$. *The average time complexity to fit each model is* $O\left(Nd+d^3\log d/\varepsilon^2\right)$.

Given two-dimensional image data $\boldsymbol{Y}$, the best order with respect to a 2D AR model is usually unknown at first. To obtain an appropriate order, a common approach is to traverse all possible orders and choose the best order according to specific criteria, such as the Akaike information criterion (Akaike, 1974; Aksasse and Radouane, 1999), the Bayesian information criterion (Schwarz, 1978). This order selection procedure involves an independent model fitting for each order, which is time-consuming when using an OLS estimator. For instance, for an input 2D AR(p,q) model with $d=|L|$ parameters, the order selection procedure involves *d* times OLS with the total time complexity $O(Nd^3)$ and the average time complexity $O(Nd^2)$ for each model. However, Algorithm 1 can obtain all model fitting results with order up to (p,q) in one operation with time complexity as in Theorem 2. To acquire all the model fitting results, we only need to use the leverage scores obtained in each iteration to fit the 2D AR model using the leverage-based subsampling method, and the precision of the fitting result is guaranteed by Theorem 1. In this approach, Algorithm 1 reduces the average time complexity for model fitting from $O(Nd^2)$ to $O\left(Nd+d^3\log d/\varepsilon^2\right)$, improving computational speed efficiently when *N* is much larger than *d*.

In practice, we recommend selecting a subsample size $s\in O(\frac{N}{d})$ to strike a balance between approximation accuracy and computational efficiency. Generally, a larger subsample size leads to more accurate parameter estimation. When *s* is less than $O(\frac{N}{d})$, the average computational cost, $O\left(Nd+sd^2\right)$, is dominated by the $O(Nd)$ term. Therefore, choosing $s\in O(\frac{N}{d})$ maximizes approximation accuracy while keeping the average time complexity at $O(Nd)$.

5 Simulation results

To evaluate the performance of the proposed subsampling method, we compare Algorithm 1 (LEV-appr) with uniform subsampling (UNIF), full-sample OLS, and the exact leverage-based subsampling method using the leverage scores calculated in Proposition 1 (LEV-exact) regarding both the estimation accuracy and computational efficiency. All the methods are implemented in `Python`.

We generate synthetic large-scale two-dimensional image data from 2D AR models with different sizes and lags, and then estimate parameters for each dataset using the methods mentioned above.

Considering causality, the images are generated sequentially from left to right and top to bottom using a causal version of the 2D AR model. In a causal 2D AR model with order (p,q), the set L in Definition 1 is replaced by $L^* = \{-p,-p+1,\ldots,0\}\times\{-q,-q+1,\ldots,0\}/\{(0,0)\}$, while all properties of the original 2D AR model are retained.

Given the order (p,q), we generate model parameters for the 2D AR model from three distinct distributions:

• `D1`: $\phi_{u,v} = (-1)^{-(u+v)}(u+v)/(p+q)\times 0.02,\ for\ (u,v)\in L^*$,

• `D2`: $\phi_{u,v} \overset{i.i.d.}{\sim} N(0,\frac{1}{25p^2q^2}),\ for\ (u,v)\in L^*$,

• `D3`: $\phi_{u,v} \overset{i.i.d.}{\sim} Unif(-\frac{1}{2(p+1)(q+1)},\frac{1}{2(p+1)(q+1)}),\ for\ (u,v)\in L^*$,

where $N(\mu,\sigma^2)$ represents normal distribution with mean μ and variance σ^2, $Unif(a,b)$ represents uniform distribution on interval (a,b).

After generating the parameters, we initialize the left *p* columns and top *q* rows of the 2D AR model and generate the remaining data as follows:

$$\begin{aligned} y_{i,j} &= \max(i,j)+\delta_{i,j},\ for\ i\le p\ or\ j\le q, \\ y_{i,j} &= \sum_{(u,v)\in L^*}\phi_{u,v}y_{i+u,j+v}+\varepsilon_{i,j},\ for\ i>p\ and\ j>q, \end{aligned} \quad (18)$$

where $\delta_{i,j} \overset{i.i.d.}{\sim} N(0,1)$ and $\varepsilon_{i,j} \overset{i.i.d.}{\sim} N(0,\sigma^2)$ with $\sigma = 0.1$.

We analyze the effect of three key factors on parameter estimation: the number of parameters, the subsample size, and the size of the time series. We adjust these factors using the following three settings:

• `S1`: Fixed time series size 1000×1000, fixed order $(5,5)$, with subsample size varying from 100 to 10000.

• S2: Fixed time series size 1000×1000, fixed subsample size 10000, with the number of parameters varying from 8 (i.e., order (2,2)) to 63 (i.e., order (7,7)).

• S3: Fixed order $(5,5)$, fixed subsample size 10000, with time series size varying from 150×150 to 1000×3000.

For each setting and each parameter, we repeat the process of data generation and parameter estimation 20 times, recording the average MSE of parameter estimation and the average time cost, respectively.

Figures 3 and 4 show the results under various settings and distributions. In Fig. 3, each row represents a particular setting (S1–S3), and each column corresponds to a specific distribution of parameters (D1–D3). The three columns in Fig. 3 (or Fig. 4) show MSE (or time cost) versus the number of parameters *d*, subsample size *s*, and time series size *N*, respectively. Note that for the proposed LEV-appr method, the subsample size *s* refers to the size used in each iteration of Algorithm 1.

Three significant observations emerge from Figs. 3 and 4. First, in Fig. 3, across all settings, the MSE of LEV-appr is nearly identical to that of LEV-exact, demonstrating the accuracy of our approximation. In contrast, Fig. 4 shows that the computational cost of LEV-appr is significantly lower than that of LEV-exact, highlighting its efficiency. Compared to uniform sampling, the proposed method consistently achieves a much smaller MSE while spending almost the same time. Furthermore, as seen in the first image of Fig. 4, the computational time of LEV-appr grows more slowly than that of full-sample OLS as the number of parameters increases. This implies that the proposed method can ensure estimation accuracy with a much lower computational cost.

Second, in the second row of Fig. 3, we observe that as the subsample size increases, the MSE of the proposed method decreases rapidly, approaching the accuracy of full-sample OLS at a subsample size of 10000. This improvement outpaces that of uniform subsampling, which struggles to match OLS accuracy even at large subsample sizes.

Third, referring to the third row of Fig. 3, the MSE of uniform subsampling increases as the time series size increases, whereas the MSE of the proposed method shows a decreasing trend. This observation suggests that the approximated leverage scores can effectively capture valuable information in large-scale datasets to enhance estimation accuracy.

6 Real data example

In this section, we apply the proposed method to two real-world tasks to demonstrate its effectiveness. We first propose utilizing standardized leverage scores of the 2D AR model as anomaly scores to distinguish anomaly pixels of an image. Subsequently, we extend this method to facial images to identify regions of skin lesions. Results showcase that our proposed method outperforms other competitors in both detection accuracy and computational efficiency.

6.1 Anomaly detection

In this section, we showcase how leverage scores of the 2D AR model can be utilized in anomaly detection tasks and outperform existing methods. We consider two hyperspectral images manually extracted from large images downloaded from the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) Web site [1], with their pseudo-grayscale representations shown in the first column of Fig. 5. The first hyperspectral image, of size $100 \times 100 \times 191$, was captured over the Gulfport airport area and has three abnormal shadows [2]. The second hyperspectral image, of size $100 \times 100 \times 189$, was taken over the San Diego airport area and contains three aircraft regarded as abnormal objects. Both images have manually obtained reference maps, on which the abnormal pixels are labeled, as shown in the second column of Fig 5.

The goal of anomaly detection is to point out anomaly pixels of hyperspectral images as accurately as possible. A typical method is to calculate an anomaly score of each pixel and choose a threshold to distinguish abnormal from normal pixels. Given that the leverage score quantifies the impact of a single data point in the model, and outliers typically exhibit high leverage scores, using leverage scores from the 2D AR model offers a natural way to assess pixel abnormality.

To apply the 2D AR model to hyperspectral images, we first average the hyperspectral cube on the third dimension to get a two-dimensional image. Then we fit a 2D AR model as in Definition 1 with order $(1,1)$ and calculate the leverage score of each pixel. Since the leverage score of one pixel is only related to its covariates, pixels around the abnormal pixel also tend to have high leverage scores. To refine this, we additionally perform a min pooling step, which replaces each pixel's leverage score with the minimum score of its neighboring pixels. This adjustment better reflects the true degree of pixel abnormality. Ultimately, the pooled leverage scores are standardized and used as anomaly scores.

To evaluate the performance of the proposed leverage approximation method (LEV-appr), we compare it with exact leverage-based subsampling (LEV-exact) and two other hyperspectral anomaly detection methods. One is the well-known Reed–Xiaoli (RX) detector (Reed and Yu, 1990), which is a classical and widely used baseline in anomaly detection experiments. The other is the state-of-the-art LARTVAD technique (Sun et al., 2023), which has demonstrated superior performance compared to most previously proposed anomaly detection methods on various datasets.

Figure 5 illustrates the anomaly scores obtained from different anomaly detection methods for two hyperspectral images. Figure 6 shows the receiver-operating characteristic (ROC) curves for anomaly detection, and Table 2 summarizes the area under the curve (AUC) and the computational time for each method.

From Fig. 5, we find that abnormal pixels have significantly higher leverage scores, making them easily distinguishable from normal pixels. Besides, the approximated leverage scores have almost the same performance as exact leverage scores. In contrast, for RX, the difference in anomaly scores between normal and abnormal pixels is not obvious. For LARTVAD, the anomaly scores of normal pixels (i.e., brightness of background) are relatively high, weakening the distinction between normal and abnormal pixels, especially in the right lower corner of the San Diego image. From Fig. 6 and Table. 2, we observe that exact and approximated leverage scores have the best anomaly detection performance while approximated leverage scores also obtain the best time efficiency. In general, anomaly

detection through leverage scores consistently outperforms its competitors in both detection accuracy and time efficiency.

6.2 Facial skin analysis

Hyperspectral imaging is gaining increasing attention in dermatology and cosmetics due to its fast and noninvasive image acquisition, no side effects, and detailed spatial and spectral information. Previous studies have demonstrated that different wavelengths of skin hyperspectral encompass diverse layers of skin information (Seroul et al., 2016; Gevaux et al., 2021). For instance, hyperspectral images can be utilized to generate maps depicting skin absorption properties like oxygen saturation, blood volume fraction, or melanin concentration through an optical-based model (Seroul et al., 2016; Gevaux et al., 2019). One fundamental task in dermatology is skin lesion area identification, which is essential and critical for subsequent analysis.

We consider a facial skin hyperspectral image dataset (Ng et al., 2023) containing 330 hyperspectral cubes from 51 subjects, each of which covers a wide range of wavelengths from the visible (VIS) spectrum (400nm–700nm) to near-infrared (NIR) spectrum (700nm–1000nm). Each hyperspectral cube has dimensions of $1024 \times 1024 \times 62$, with 31 bands for VIS, and 31 bands for NIR.

We propose identifying skin lesion regions through leverage scores of the 2D AR model. We select three skin regions simultaneously containing pigmentation and acne, which can be seen clearly from the visible light RGB pseudo-color images. We focus on the wavelength band around 900nm for these regions and calculate the leverage scores of corresponding 2D AR models, using both the exact formulation and Algorithm 1. The results are compared with the RX and LARTVAD anomaly detectors mentioned in Section 6.1.

Figure 7 displays the pseudo-color images and anomaly scores for the three skin regions. Both pigmentation and acne can be seen clearly in pseudo-color images. However, due to strong noise, the RX detector and LARTVAD fail to perform effectively. In contrast, the leverage scores in the forth and fifth columns imply the abnormal degree of each pixel when fitting a 2D AR model on the 900nm wavelength band of hyperspectral cubes. Acne is emphasized by high leverage scores, while pigmentation, despite noticeable in the pseudo-color images, appears less prominent. In this manner, acne can be efficiently identified by selecting pixels with leverage scores above a certain threshold. Additionally, leverage scores approximated by Algorithm 1 closely match those obtained from the exact method, further validating the effectiveness of Algorithm 1.

7 Discussion

In this paper, we propose a novel approach to accelerate leverage score computation and parameter estimation in the 2D AR model. Specifically, we derive a recursive formula for leverage scores in 2D AR models and develop an efficient algorithm to approximate them. Theoretically, we establish an error bound for these approximations and analyze the time complexity for the proposed method. Extensive experiments on synthetic datasets demonstrate the accuracy and efficiency of our approach. Moreover, we apply the method to an anomaly detection dataset and a hyperspectral skin dataset, showing its practical utility in achieving competitive results with significantly less computational burden.

For both anomaly detection and facial skin analysis, although our approach outperforms competitors in terms of detection performance and computational efficiency, it relies on the fact that information of these hyperspectral can be compressed into a two-dimensional image. In the case of anomaly detection, anomaly pixels still exhibit anomalies on images obtained by averaging the hyperspectral data along the third dimension. For facial skin analysis, we select NIR channels—guided by prior knowledge—that are particularly effective in distinguishing acne from pigmentation. Without these prerequisites, the anomaly detection performance using the leverage scores of the 2D AR model would be reduced. This limitation is inherent to the 2D AR model, which is designed solely for processing two-dimensional images.

Three-dimensional (3D) AR models and two-dimensional (2D) vector autoregressive (VAR) models offer a promising path to overcoming the inherent limitations of the 2D AR model and effectively processing 3D data. Both the 3D AR and 2D VAR models can be naturally applied to 3D datasets, capturing significantly richer information than the 2D AR model. For example, when dealing with hyperspectral data, fitting a 3D AR model enables us to obtain leverage scores for each pixel across all channels, thereby pinpointing both the spatial location and the specific wavelength of an anomaly, which aids in determining both its position and type. For the 2D VAR model, each pixel is treated as a vector containing information from all channels, making its leverage scores highly responsive to any abnormal shifts and thereby enabling precise anomaly detection. However, the huge computational burden imposed by the large number of parameters greatly limits the practicality of these two models. Our future work will mainly focus on improving the computational efficiency of fitting the 3D AR model and the 2D VAR model, aiming to apply them effectively in practical data processing scenarios.

SUPPLEMENTARY MATERIAL

Appendix: Contains complete proofs of the theoretical results of the proposed method. Contains sections discussing order selection and hypothesis test for the proposed method. (appendix.pdf, a pdf file)

Code: Contains code that implements the proposed method and reproduces the numerical results. A readme file describing the contents is included. (code.zip, a zip file)

Acknowledgment

This work is supported by Beijing Municipal Natural Science Foundation No. 1232019, National Natural Science Foundation of China Grant No. 12301381.

Disclosure Statement

The authors report there are no competing interests to declare.

References

Ai, M., Wang, F., Yu, J., and Zhang, H. (2021), “Optimal subsampling for large-scale quantile regression,” *Journal of Complexity*, 62, 101512.

Akaike, H. (1974), "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, 19, 716–723.

Aksasse, B. and Radouane, L. (1999), "Two-dimensional autoregressive (2-D AR) model order estimation," *IEEE Transactions on Signal Processing*, 47, 2072–2077.

Choi, B. and Politis, D. N. (2007), "Modeling 2-D AR Processes With Various Regions of Support," *IEEE Transactions on Signal Processing*, 55, 1696–1707.

Derezinski, M. and Warmuth, M. K. K. (2017), "Unbiased estimates for linear regression via volume sampling," in *Advances in Neural Information Processing Systems*, volume 30.

Derezinski, M., Warmuth, M. K. K., and Hsu, D. J. (2018), ''Leveraged volume sampling for linear regression," in *Advances in Neural Information Processing Systems*, volume 31.

Drineas, P., Magdon-Ismail, M., Mahoney, M. W., and Woodruff, D. P. (2012), "Fast approximation of matrix coherence and statistical leverage," *The Journal of Machine Learning Research*, 13, 3475–3506.

Drineas, P., Mahoney, M. W., and Muthukrishnan, S. (2006), "Sampling algorithms for l_2 regression and applications," in *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*.

Drineas, P., Mahoney, M. W., Muthukrishnan, S., and Sarlós, T. (2011), "Faster least squares approximation," *Numerische mathematik*, 117, 219–249.

Eshragh, A., Roosta, F., Nazari, A., and Mahoney, M. W. (2022), ''LSAR: efficient leverage score sampling algorithm for the analysis of big time series data," *Journal of Machine Learning Research*, 23, 1–36.

Feldman, D., Faulkner, M., and Krause, A. (2011), "Scalable training of mixture models via coresets," in *Advances in Neural Information Processing Systems*.

Gevaux, L., Adnet, C., Séroul, P., Clerc, R., Trémeau, A., Perrot, J. L., and Hébert, M. (2019), "Three-dimensional maps of human skin properties on full face with shadows using 3-D hyperspectral imaging," *Journal of Biomedical Optics*, 24, 066002.

Gevaux, L., Gierschendorf, J., Rengot, J., Cherel, M., Séroul, P., Nkengne, A., Robic, J., Trémeau, A., and Hébert, M. (2021), "Real-time skin chromophore estimation from hyperspectral images using a neural network," *Skin Research and Technology*, 27, 163–177.

Gu, C. and Kim, Y.-J. (2002), "Penalized Likelihood Regression: General Formulation and Efficient Approximation," *Canadian Journal of Statistics*, 30, 619–628.

Hu, Y., Li, M., Liu, X., and Meng, C. (2025), "Sampling-based methods for multi-block optimization problems over transport polytopes," *Mathematics of Computation*.

Jang, J., Oh, S., Kim, Y., Seo, D., Choi, Y., and Yang, H. J. (2023), "M^2SODAI: Multi-Modal Maritime Object Detection Dataset With RGB and Hyperspectral Image Sensors," in *Advances in Neural Information Processing Systems*, volume 36.

Jun Yu, HaiYing Wang, M. A. and Zhang, H. (2022), "Optimal Distributed Subsampling for Maximum Quasi-Likelihood Estimators With Massive Data," *Journal of the American Statistical Association*, 117, 265–276.

Kang, X., Zhang, X., Li, S., Li, K., Li, J., and Benediktsson, J. A. (2017), "Hyperspectral Anomaly Detection With Attribute and Edge-Preserving Filters," *IEEE Transactions on Geoscience and Remote Sensing*, 55, 5600–5611.

Kaufman, H. and Tekalp, A. (1991), "Survey of estimation techniques in image restoration," *IEEE Control Systems Magazine*, 11, 16–24.

Kokaram, A. (2004), "A statistical framework for picture reconstruction using 2D AR models," *Image and Vision Computing*, 22, 165–171. Statistical Methods in Video Processing.

Köppel, M., Doshkov, D., Racape, F., Ndjiki-Nya, P., and Wiegand, T. (2015), "On the usage of the 2D-AR-model in texture completion scenarios with causal boundary conditions: A tutorial," *Signal Processing: Image Communication*, 32, 106–120.

Li, M., Yu, J., Li, T., and Meng, C. (2023a), "Importance Sparsification for Sinkhorn Algorithm," *Journal of Machine Learning Research*, 24, 1–44.

— (2024a), "Core-Elements for Large-Scale Least Squares Estimation," *Statistics and Computing*, 34, 1–16.

Li, M., Yu, J., Xu, H., and and, C. M. (2023b), "Efficient Approximation of Gromov-Wasserstein Distance Using Importance Sparsification," *Journal of Computational and Graphical Statistics*, 32, 1512–1523.

Li, M., Zhang, J., and Meng, C. (2024b), "Nonparametric Additive Models for Billion Observations," *Journal of Computational and Graphical Statistics*, 1–16.

Li, T. and Meng, C. (2020), "Modern subsampling methods for large-scale least squares regression," *International Journal of Cyber-Physical Systems*, 2, 1–28.

Lu, S. W. and Xu, H. (1995), "Textured image segmentation using autoregressive model and artificial neural network," *Pattern Recognition*, 28, 1807–1817.

Ma, P., Chen, Y., Zhang, X., Xing, X., Ma, J., and Mahoney, M. W. (2022), "Asymptotic Analysis of Sampling Estimators for Randomized Numerical Linear Algebra Algorithms," *The Journal of Machine Learning Research*, 23, 1–45.

Ma, P., Mahoney, M. W., and Yu, B. (2015), "A statistical perspective on algorithmic leveraging," *The Journal of Machine Learning Research*, 16, 861–911.

Ma, P. and Sun, X. (2015), "Leveraging for big data regression," *Wiley Interdisciplinary Reviews: Computational Statistics*, 7, 70–76.

Meng, C., Wang, Y., Zhang, X., Mandal, A., Zhong, W., and Ma, P. (2017), "Effective statistical methods for big data analytics," in *Handbook of research on applied cybernetics and systems science*, IGI Global, 280–299.

Meng, C., Xie, R., Mandal, A., Zhang, X., Zhong, W., and Ma, P. (2021), "LowCon: A Design-based Subsampling Approach in a Misspecified Linear Model," *Journal of Computational and Graphical Statistics*, 30, 694–708.

Meng, C., Yu, J., Chen, Y., Zhong, W., and Ma, P. (2022), "Smoothing splines approximation using Hilbert curve basis selection," *Journal of Computational and Graphical Statistics*, 31, 802–812.

Meng, C., Zhang, X., Zhang, J., Zhong, W., and Ma, P. (2020), "More efficient approximation of smoothing splines via space-filling basis selection," *Biometrika*, 107, 723–735.

Ng, P. C., Chi, Z., Verdie, Y., Lu, J., and Plataniotis, K. N. (2023), "Hyper-Skin: A Hyperspectral Dataset for Reconstructing Facial Skin-Spectra from RGB Images," in *Advances in Neural Information Processing Systems*, volume 36.

Oe, S. (1993), "Texture segmentation method by using two-dimensional AR model and Kullback information," *Pattern Recognition*, 26, 237–244.

Palm, B. G., Bayer, F. M., and Cintra, R. J. (2022), "2-D Rayleigh autoregressive moving average model for SAR image modeling," *Computational Statistics & Data Analysis*, 171, 107453.

Pukelsheim, F. (2006), *Optimal Design of Experiments*, Society for Industrial and Applied Mathematics.

Reed, I. and Yu, X. (1990), "Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38, 1760–1770.

Schwarz, G. (1978), "Estimating the Dimension of a Model," *The Annals of Statistics*, 6, 461–464.

Seroul, P., Hébert, M., Chérel, M., Vernet, R., Clerc, R., and Jomier, M. (2016), "Model-based skin pigment cartography by high-resolution hyperspectral imaging," *Journal of Electronic Imaging*, 69, 48–56.

Sun, S., Liu, J., Chen, X., Li, W., and Li, H. (2023), "Hyperspectral Anomaly Detection With Tensor Average Rank and Piecewise Smoothness Constraints," *IEEE Transactions on Neural Networks and Learning Systems*, 34, 8679–8692.

Wang, H., Yang, M., and Stufken, J. (2019), "Information-Based Optimal Subdata Selection for Big Data Linear Regression," *Journal of the American Statistical Association*, 114, 393–405.

Wang, H., Zhu, R., and Ma, P. (2018), ‘‘Optimal subsampling for large sample logistic regression,” *Journal of the American Statistical Association,* 113, 829–844.

Wang, Y., Yu, A. W., and Singh, A. (2017), “On Computationally Tractable Selection of Experiments in Measurement-Constrained Regression Models,” *Journal of Machine Learning Research,* 18, 1–41.

Xie, R., Bai, S., and Ma, P. (2023), “Optimal sampling designs for multidimensional streaming time series with application to power grid sensor data,” *The Annals of Applied Statistics,* 17, 3195 – 3215.

Xie, R., Wang, Z., Bai, S., Ma, P., and Zhong, W. (2019), “Online Decentralized Leverage Score Sampling for Streaming Multidimensional Time Series,” in *The 22nd International Conference on Artificial Intelligence and Statistics*.

Zhang, D. and Liang, J. (2017), ‘‘Graph-Based Transform for 2D Piecewise Smooth Signals With Random Discontinuity Locations,” *IEEE Transactions on Image Processing,* 26, 1679–1693.

Zhang, X., Xie, R., and Ma, P. (2018), “Statistical Leveraging Methods in Big Data,” in *Handbook of Big Data Analytics,* Springer, 51–74.

Zielinski, J., Bouaynaya, N., and Schonfeld, D. (2010), “Two-dimensional ARMA modeling for breast cancer detection and classification,” in *2010 International Conference on Signal Processing and Communications (SPCOM)*.

Notes

[1] http://aviris.jpl.nasa.gov/

[2] The dataset is available on the first author’s homepage: http://xudongkang.weebly.com/

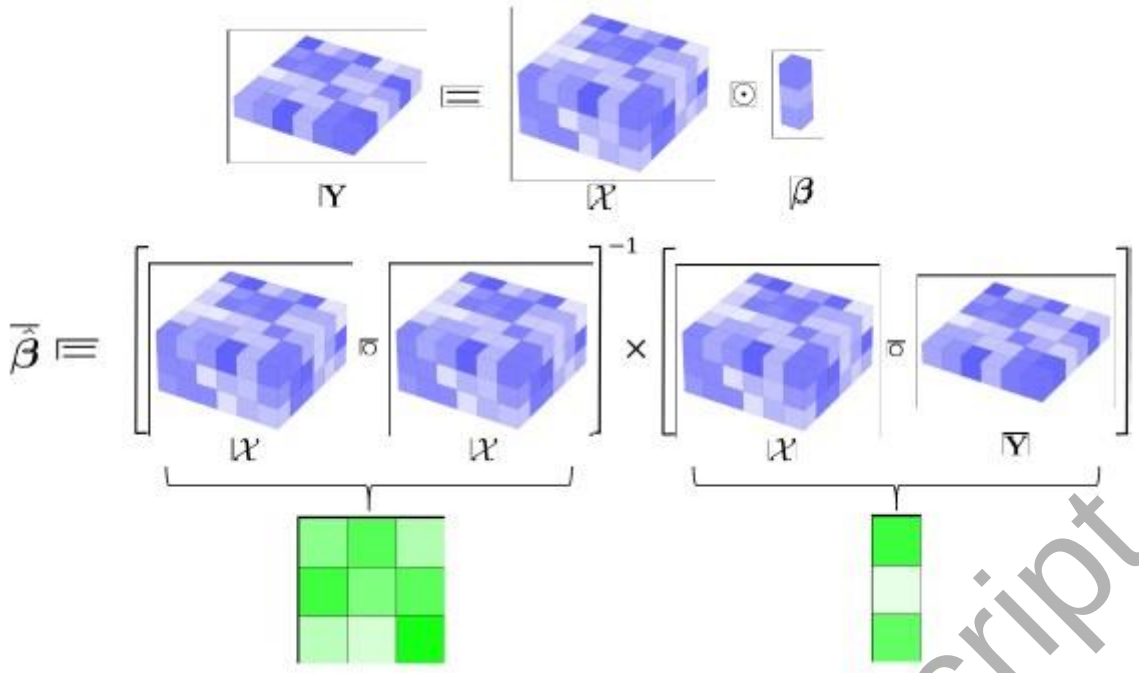

Figure 1: An illustration of the 2D AR model in tensor form. Different colors represent different values in tensor(matrix, vector).

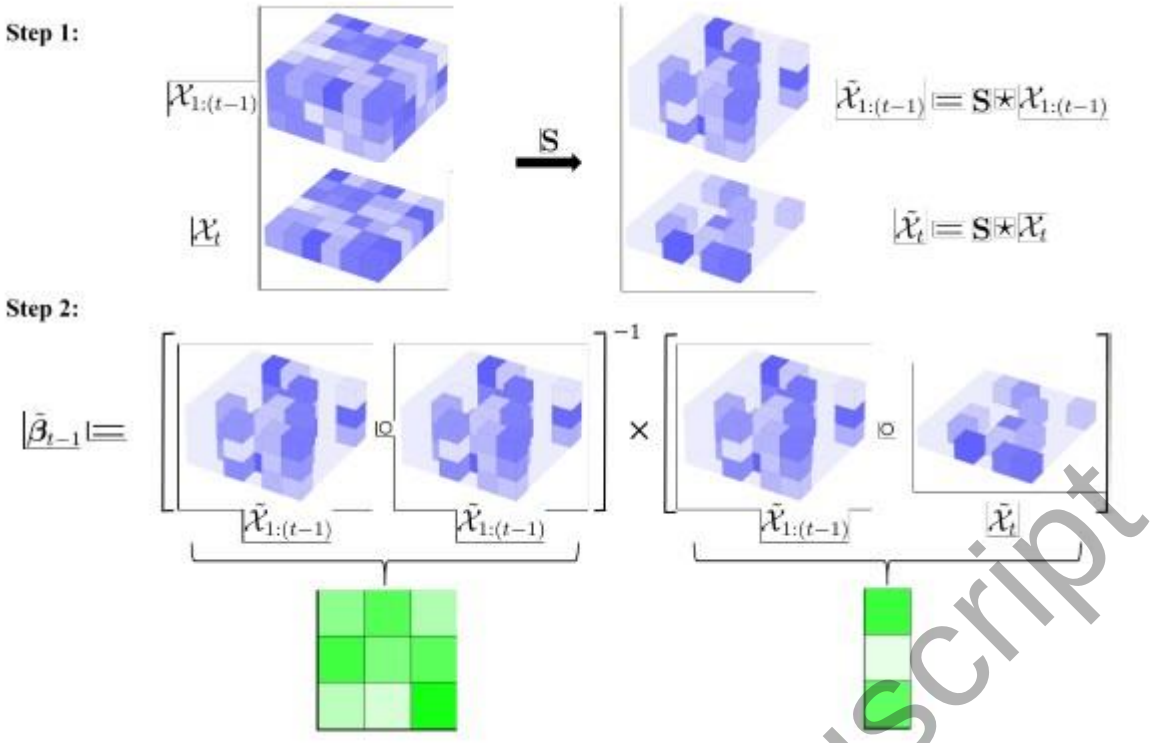

Figure 2: An illustration of parameter approximation by subsampling. The non-zero elements of each tensor(matrix, vector) are labeled with non-transport colors. Step 1 illustrates the subsampling process through mask $\mathbf{S}$. Step 2 illustrates the parameter approximation.

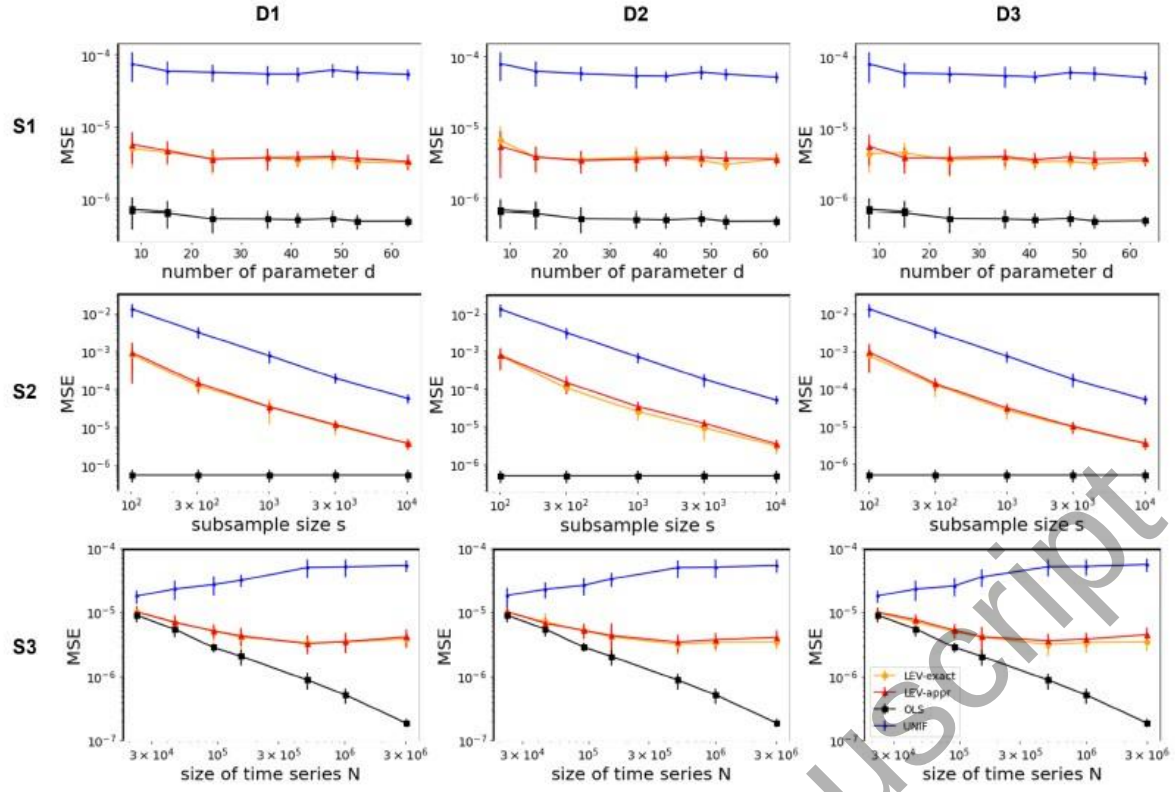

Figure 3: Comparison of MSE under different settings (rows) and distributions (columns). MSE is plotted versus the number of parameters, subsample size, and time series size, respectively. Vertical bars represent the standard errors.

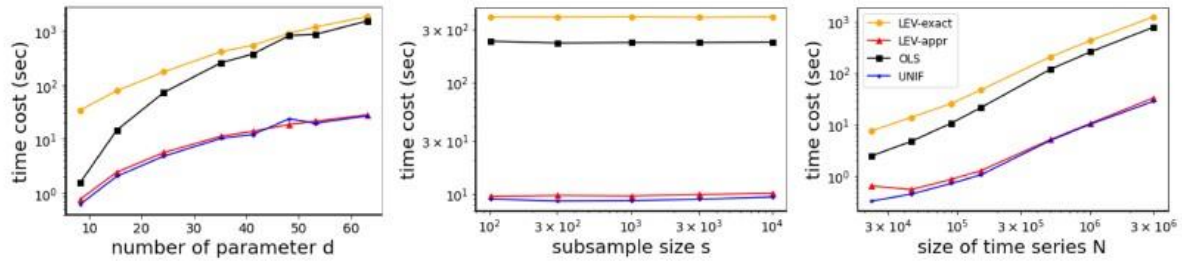

Figure 4: Total time cost of order selection and parameter estimation versus the number of parameters, subsample size, and time series size, respectively.

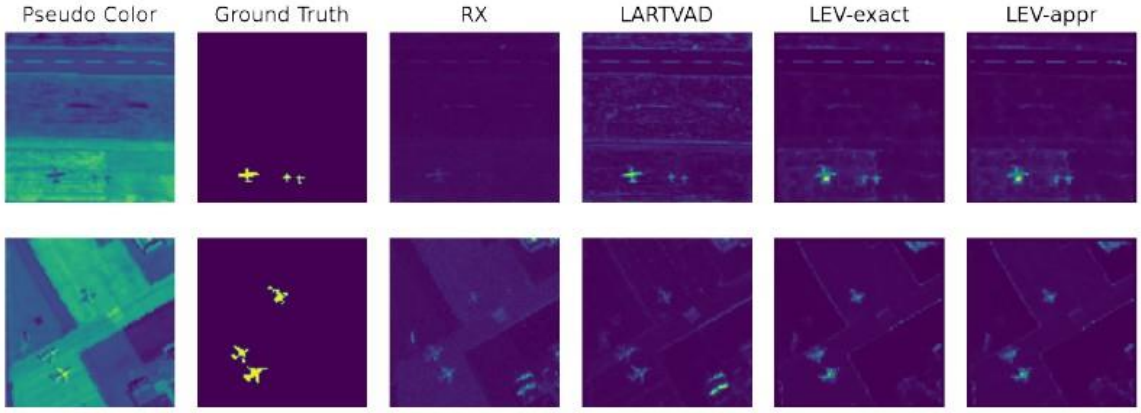

Figure 5: Two hyperspectral images and their detection maps obtained by different methods. The top row is Gulfport, and the bottom row is San Diego.

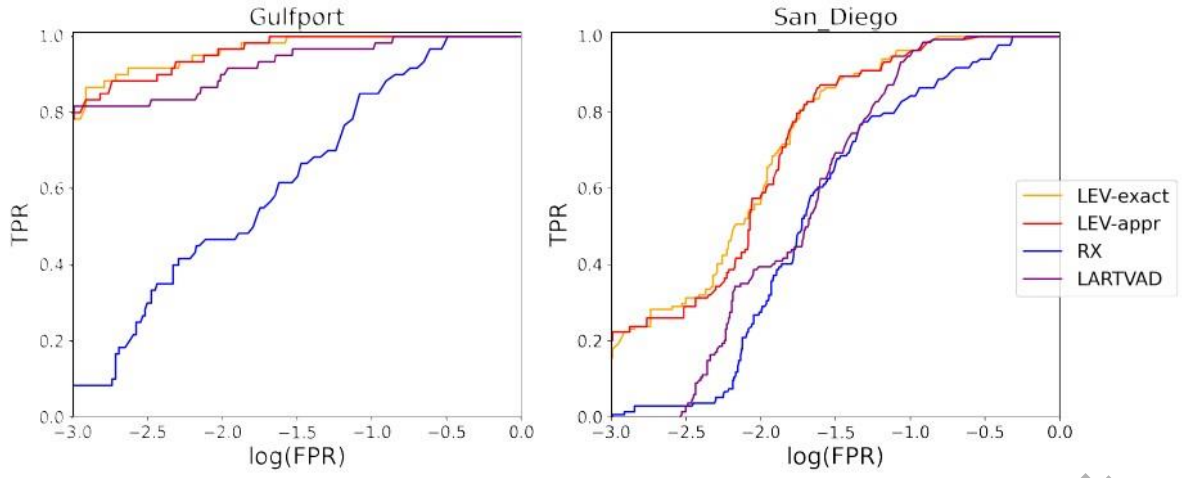

Figure 6: ROC curves of different methods on two hyperspectral images.

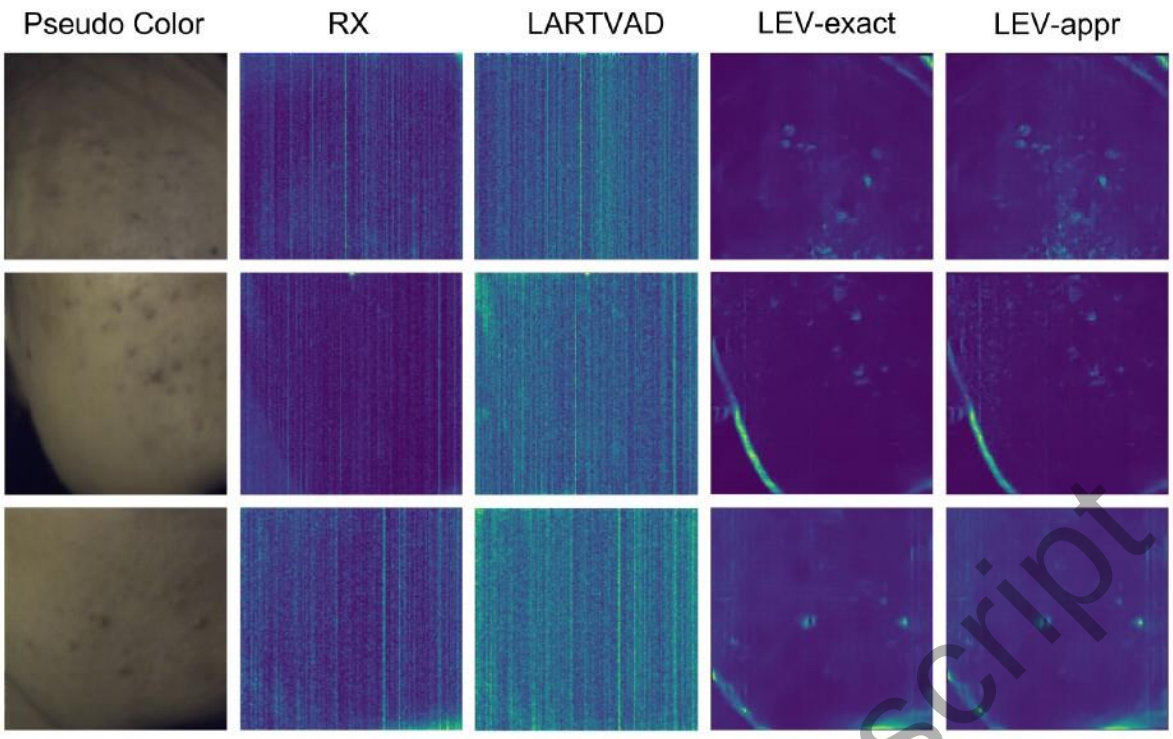

Figure 7: Pseudo-color and anomaly scores of three skin regions. Each row corresponds to a specific skin region. The first column shows the pseudo-color images from the VIS hyperspectral cubes. The remaining columns display the anomaly scores obtained from RX, LARTVAD, LEV-exact, and LEV-appr, respectively.

Table 1: Summary of tensor and matrix operations, where $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{m \times n}$, $\mathcal{X} \in \mathbb{R}^{m \times n \times c_1}, \mathcal{Y} \in \mathbb{R}^{m \times n \times c_2}$, and $\mathbf{y} \in \mathbb{R}^{c_1}$.

Operation	Notation and description
Frobenius inner product	$\langle \mathbf{X}, \mathbf{Y} \rangle_F := \sum_{i=1}^{m} \sum_{j=1}^{n} X_{i,j} Y_{i,j}$, $\| \mathbf{X} \|_F^2 := \langle \mathbf{X}, \mathbf{X} \rangle_F$
Matrix element-wise product	$\mathbf{X} \star \mathbf{Y} := \{ \mathbf{M} \in \mathbb{R}^{m \times n} \mid M_{i,j} = X_{i,j} Y_{i,j}, i \in [m], j \in [n] \}$
Tensor-matrix element-wise product	$\mathbf{X} \star \mathcal{Y} := \{ \mathcal{M} \in \mathbb{R}^{m \times n \times c_2} \mid \mathcal{M}(i, j, :) = X_{i,j} \mathcal{Y}(i, j, :), i \in [m], j \in [n] \}$
Tensor-matrix product	$\mathcal{X} \circ \mathbf{Y} := \{ \mathbf{a} \in \mathbb{R}^{c_1} \mid a_i = \langle \mathcal{X}(:, :, i), \mathbf{Y} \rangle_F, i \in [c_1] \}$
Tensor-tensor product	$\mathcal{X} \circ \mathcal{Y} := \{ \mathbf{A} \in \mathbb{R}^{c_1 \times c_2} \mid A_{i,j} = \langle \mathcal{X}(:, :, i), \mathcal{Y}(:, :, j) \rangle_F, i \in [c_1], j \in [c_2] \}$
Tensor-vector product	$\mathcal{X} \odot \mathbf{y} := \{ \mathbf{A} \in \mathbb{R}^{m \times n} \mid A_{i,j} = \langle \mathcal{X}(i, j, :), \mathbf{y} \rangle, i \in [m], j \in [n] \}$

Table 2: AUC and time cost (seconds) of different methods on two hyperspectral images.

		RX	LARTVAD	LEV-exact	LEV-appr
AUC	Gulfport	0.950	**0.994**	**0.998**	**0.998**
	San Diego	0.939	0.969	**0.983**	**0.981**
Time cost	Gulfport	4.021	26.322	0.243	**0.050**
	San Diego	3.895	26.543	0.304	**0.086**